

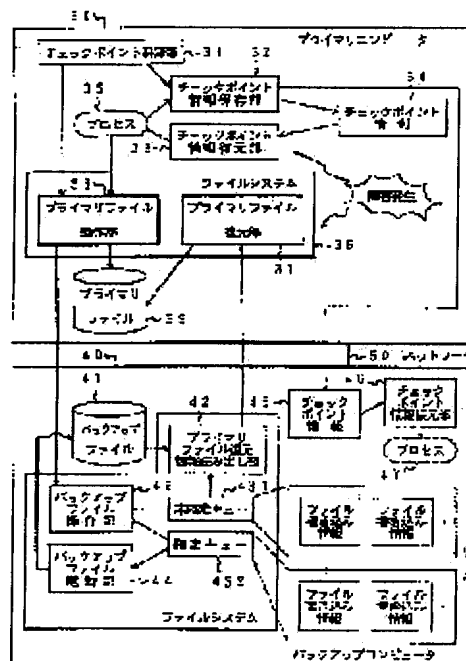
(11)Publication number : **10-133927**  
(43)Date of publication of application : **22.05.1998**

G06F 12/00

(21)Application number : **09-232930** (71)Applicant : **TOSHIBA CORP**  
(22)Date of filing : **28.08.1997** (72)Inventor : **HIRAYAMA HIDEAKI**  
**SHIROKIBARA TOSHIO**

Priority number : **08233021**      Priority date : **03.09.1996**      Priority country : **JP**

**SOLUTION:** When writing, etc., is requested to the file, 'file writing information' is preserved in an unidentified queue 431 to instantly update only a primary file 39. Then after a check point is picked up, 'file writing information' preserved in the queue 431 is moved to an identified queue 432 to reflect to a backup file 41. On the other hand, at the time of recovery, all the pieces of data before update corresponding to data updated after a finally picked check point are read from the file 41 based on 'file writing information' preserved in the queue 431 to recover the file 39 to the point of a check point time by us



(19)日本国特許庁 (J P)

(12) 公 開 特 許 公 報 (A)

(11)特許出願公開番号

特開平10-133927

(43)公開日 平成10年(1998) 5月22日

(51)Int.Cl.<sup>8</sup>  
G 0 6 F 12/00

識別記号  
5 3 1

F I  
C 0 6 F 12/00

5 3 1 R  
5 3 1 M

審査請求 未請求 請求項の数16 O L (全 15 頁)

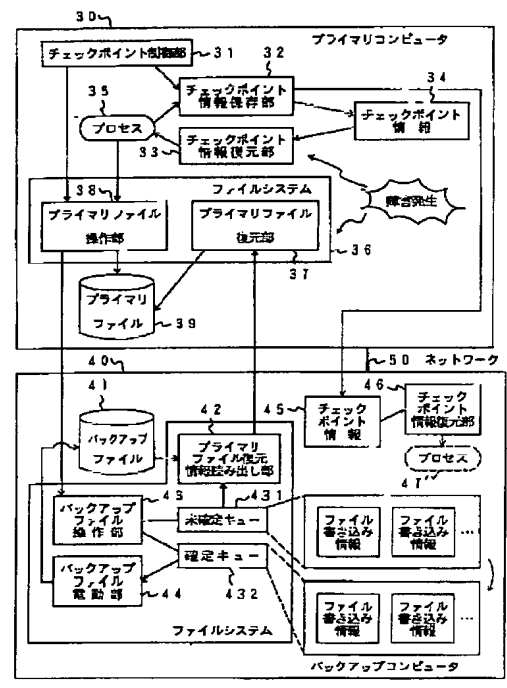
(21)出願番号 特願平9-732930  
(22)出願日 平成9年(1997) 8月28日  
(31)優先権主張番号 特願平8-733021  
(32)優先日 平8 (1996) 9月3日  
(33)優先権主張国 日本 (J P)

(71)出願人 000003078  
株式会社東芝  
神奈川県川崎市幸区堀川町72番地  
(72)発明者 平山 秀昭  
東京都青梅市末広町2丁目9番地 株式会  
社東芝青梅工場内  
(72)発明者 白木原 敏雄  
神奈川県川崎市幸区小向東芝町1番地 株  
式会社東芝研究開発センター内  
(74)代理人 弁理士 鈴江 武彦 (外6名)

(54)【発明の名称】 コンピュータシステムおよびファイル管理方法

(57)【要約】

【課題】ファイルを更新する際に更新前データの退避を待機することなく障害発生時のロールバックを実現するコンピュータシステムを提供する。  
【解決手段】ファイルに対して書き込みなどが要求された際、その「ファイル書き込み情報」を未確定キュー431に保存してプライマリファイル39のみを即座に更新する。そして、チェックポイントが採取された後に、未確定キュー431に保存された「ファイル書き込み情報」を確定キュー432に移動させてバックアップファイル41への反映を行なう。一方、リカバリを行なうときには、未確定キュー431に保存された「ファイル書き込み情報」に基づき、最後に採取したチェックポイント以降に更新されたデータに対応する更新前のデータをバックアップファイル41からすべて読み出し、この読み出した更新前のデータを用いてプライマリファイル39をチェックポイント時点に復元する。



(2)

特開平10-133927

**【特許請求の範囲】**

【請求項1】 運用系および待機系の2つのコンピュータで2重化されたコンピュータシステムであって、中断された処理を再開するためのチェックポイントを定期的に採取し、前記運用系および待機系双方のコンピュータ上に保存するコンピュータシステムにおいて、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータで2重化して設けておき、前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知する手段と、前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させる手段とを具備してなることを特徴とするコンピュータシステム。

【請求項2】 前記更新情報を前記運用系のコンピュータ上にバッファリングしておき、前記チェックポイントの採取時点までに前記待機系のコンピュータに一括転送する手段をさらに具備してなることを特徴とする請求項1記載のコンピュータシステム。

【請求項3】 前記プロセスがアボートしたときに、最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項4】 前記プロセスがアボートしたときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項5】 前記運用系のコンピュータまたはこの運用系のコンピュータを制御するオペレーティングシステムに障害が発生したときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項6】 前記待機系のコンピュータまたはこの待機系のコンピュータを制御するオペレーティングシステムに障害が発生したときに、前記チェックポイントおよび更新情報の待機系のコンピュータへの転送を停止する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項7】 前記運用系のファイルに障害が発生したときに、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項8】 前記待機系のファイルに障害が発生したときに、前記チェックポイントおよび更新情報の待機系のコンピュータへの転送を停止する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項9】 待機系のファイルの切り離しが行なわれたときに、第3のコンピュータ上に新たに待機系のファイルを確認する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項10】 待機系のファイルを用いて前記プロセスの前記チェックポイントからの再実行が行なわれたときに、前記待機系のファイルを運用系に切り替えて、前記運用系のコンピュータ上に新たに待機系のファイルを確認する手段をさらに具備してなることを特徴とする請求項1または2記載のコンピュータシステム。

【請求項11】 運用系および待機系の2つのコンピュータで2重化され、中断された処理を再開するためのチェックポイントを定期的に採取して前記運用系および待機系双方のコンピュータ上に保存し、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータ上で2重化して設けたコンピュータシステムのファイル管理方法において、前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知するステップと、前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させるステップとを具備してなることを特徴とするファイル管理方法。

【請求項12】 最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求項11記載のコンピュータシステム。

【請求項13】 最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求項12記載のコンピュータシステム。

【請求項14】 最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求項13記載のコンピュータシステム。

(3)

特開平10-133927

項11記載のファイル管理方法。

【請求項13】最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行するステップをさらに備えたことを特徴とする請求項11記載のファイル管理方法。

【請求項14】運用系および待機系の2つのコンピュータで2重化され、中断された処理を再開するためのチェックポイントを定期的に採取して前記運用系および待機系双方のコンピュータ上に保存し、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータ上で多重化して設けたコンピュータシステムのファイルを管理するためのプログラムであって、

前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対して更新完了を通知し、前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させるように前記コンピュータを動作させるためのプログラムを格納したコンピュータ読取可能な記憶媒体。

【請求項15】前記プログラムは、最後のチェックポイント以降に実行されたファイルの更新に対する更新前のデータを前記更新情報により前記待機系のファイルから読み出し、前記運用系のファイルを前記チェックポイント時点の状態に復元した後、前記プロセスを前記チェックポイントから再実行するように前記コンピュータをさらに動作させる請求項14記載のコンピュータ読取可能な記憶媒体。

【請求項16】前記プログラムは、最後のチェックポイント以降に保存された更新情報を削除し、前記チェックポイント以前の更新情報により示される更新を前記待機系のファイルに反映させた後、前記プロセスを前記待機系のコンピュータ上で前記チェックポイントから再実行するように前記コンピュータをさらに動作させる請求項14記載のコンピュータ読取可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、たとえばネットワーク接続された複数のコンピュータにより構成されるネットワークコンピューティング環境などにおいて、高い信頼性を必要とするグループコンピューティング処理、データベース処理、およびトランザクション処理などに適用して好適なコンピュータシステムおよびファイル管理方法に関する。

【0002】

【従来の技術】CPUによって実行されるプロセスのア

ドレス空間やコンテキスト、およびファイルなどの状態を定期的に採取して（これをチェックポイントと称す）、障害が発生したときに、最後に採取したチェックポイントの状態を復元し、その時点からプロセスの実行を再開するといった障害からの回復機能を有したシステムにおいては、従来より外部入出力処理に関して課題があった。すなわち、障害が発生したときに、最後に採取したチェックポイントからプロセスを再実行させる際、プロセスのアドレス空間やプロセッサのコンテキストなどの状態は復元できるが、外部入力装置の状態の復元は容易ではなかった。

【0003】たとえば、ファイルに対する書き込みをキャンセルすることは困難であるために、ファイルに対して書き込みを行なうときには、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後にファイルへのデータ書き込みを行なっていた。

【0004】図15は、ファイルに対する書き込みをキャンセルすることが困難なため、ファイルに対して書き込みを行なうときに、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後にファイルへのデータ書き込みを行なう従来のシステムの仕組みを説明する図である。

【0005】この例では、時刻t1のチェックポイントを採取した時点において“ABCD”の4バイトのデータからなるファイルに、時刻t2において1バイト目に“X”をwriteしている(1)。この場合、従来では、ファイルの1バイト目に“X”をwriteする前に、ファイルの1バイト目のデータ“B”をreadしておき（これをundoログとも言う）(2)、その後でファイルの1バイト目に“X”をwriteしている(3)。

【0006】その後、時刻t3において障害が発生したために、プロセスを最後に採取されたチェックポイントの状態(t1)にロールバックする。ファイルは、チェックポイントt1以降に1バイト目が“X”に更新されているが、更新時に採取されたundoログを用いることにより、チェックポイントt1の状態を復元している。なお、このundoログは、次のチェックポイント時に不要となり廃棄される。

【0007】また、たとえば2つのコンピュータにより構築され、その一方（プライマリコンピュータ）を運用系、他方（バックアップコンピュータ）を待機系として振り分けて2重化し、プライマリコンピュータに障害が発生したときに、バックアップコンピュータが処理を引き継ぐことによってシステムの可用性を高めるといったシステムも存在する。そして、このようなシステムで、前述したようにチェックポイントを定期的に採取していけば、信頼性をさらに向上させることが可能となる。

【0008】

(4)

特開平10-133927

【発明が解決しようとする課題】この様に、プロセスのアドレス空間やコンテキスト、およびファイルなどの状態、すなわち、チェックポイントを定期的に採取している、障害が発生したときに、最後に採取したチェックポイントの状態を復元し、その時点からプロセスの実行を再開するといった障害からの回復機能を有したシステム（2重化されているかどうかを問わない）においては、その信頼性は向上されるが、一方で、ファイルの更新（たとえば書き込み）を行なうときに、一旦更新前のデータをファイルから読み込んで、それからファイルへの更新を行なわなければならないかったために、ファイルの更新性能を低下させるという課題があった。

【0009】この発明は、このような実情に鑑みてなされたものであり、チェックポイントを定期的に採取して、障害が発生したときには最後に採取したチェックポイントの状況を復元し、その時点からプロセスの実行を再開するといった障害からの回復機能を有したシステムにおいて、ファイルの更新を行なうときに、更新前のデータをファイルから読み込むなどといったことを不要とし、ファイルの更新性能を大幅に改善することを可能とするコンピュータシステムおよびファイル管理方法を提供することを目的とする。

【0010】

【課題を解決するための手段】この発明のコンピュータシステムは、運用系および待機系の2つのコンピュータで2重化されたコンピュータシステムであって、中断された処理を再開するためのチェックポイントを定期的に採取し、前記運用系および待機系双方のコンピュータ上に保存するコンピュータシステムにおいて、前記運用系のコンピュータ上で実行されるプロセスによって更新されるファイルを前記運用系および待機系双方のコンピュータで2重化して設けておき、前記プロセスからファイルの更新が指示されたときに、その更新情報を前記待機系のコンピュータ上に保存して運用系のファイルのみを更新し、その更新が完了した時点でその更新の要求元に対し更新完了を通知する手段と、前記チェックポイントが採取された後に、前記更新情報に示される更新内容を前記待機系のファイルに反映させる手段とを具備してなることを特徴とする。

【0011】この発明のコンピュータシステムにおいては、プロセスがファイルの更新を要求したときに、その更新内容を示す更新情報を取得して保存するとともに、運用系のコンピュータに配置されたファイル（運用系ファイル）のみを即座に更新して、その結果を要求元であるプロセスに返答する。そして、チェックポイントが採取された後に、その保存しておいた更新情報で示される更新内容を、待機系のコンピュータに配置されたファイル（待機系ファイル）に反映させる。

【0012】一方、たとえばプロセスがアボートしたときなどには、保存しておいた更新情報に基づいて、最後

に採取したチェックポイント以降に更新されたデータに対応する更新前のデータを待機系ファイルからすべて読み出し、この読み出した更新前のデータを用いて運用系ファイルをチェックポイント時点で復元する。

【0013】すなわち、このコンピュータシステムにおいては、従来のようにファイルを更新するときに、更新前のデータを読み出して退避させておくといった処理の完了を通常処理に待機させることなく、障害時のファイルのリカバリが実現されることになり、信頼性を損なうことなくファイルの更新性能を飛躍的に向上させることが可能となる。

【0014】また、運用系ファイルの復元に代えて、最終のチェックポイント以前に保存された更新情報で示される更新内容すべてが反映された待機系ファイルを用いたチェックポイントからのプロセスの再実行も有効である。すなわち、運用系のコンピュータの障害などにより、運用系ファイルを用いての再開が不可能な場合などにおける処理の継続も確保されることになり、システムの可用性を向上させることになる。また、この場合には、第3のコンピュータに新たに待機系ファイルを確保すれば、システムの可用性をさらに向上させることが可能となる。

【0015】

【発明の実施の形態】まず、図1を参照してこの発明の基本原理解を説明する。図1に示すように、この発明のコンピュータシステムは、運用系システム10と待機系システム20とで多重化されたシステムを前提とする。以下にそれぞれの動作を説明する。

【0016】（通常処理）

（1）運用系システム10でアプリケーションプログラム11がWriteシステムコールを発行する。

【0017】（2）ジャケットルーチン12がWriteシステムコールをフックし、運用系のオペレーティングシステムにWriteシステムコールを発行するとともに、そのWrite要求を待機系システム20に送信する。ただし、待機系システム20に即座にWrite要求を送信する必要はなく、次のチェックポイントまでに送信すればよい。また、待機系システム20では、受信したWrite要求を即座に実行するのではなく、一旦未確定キュー211に格納する。

【0018】（3）チェックポイント処理が指示されると、運用系システム10は、溜っているWrite要求をすべて待機系システム20に送信し終えなければならない。

【0019】（4）一方、待機系システム20では、未確定キュー211に格納されたWrite要求を確定キュー212に移動する。

【0020】（5）この確定キュー212に移されたWrite要求は、待機系システム20のオペレーティングシステムによって順次処理されていく。

(5)

特開平10-133927

【0021】すなわち、通常処理において発生するファイル更新においては、更新前のデータを読み出して退避させておくといった処理の完了を待機することがない。

【0022】(ロールバック処理)

(3) 障害が発生したようなときに、運用系システム10および待機系システム20の双方にロールバック処理が指示される。

【0023】(4) このとき、運用系システム10に残存するWrite要求を、すべて待機系システム20に送信する。また、待機系の未確定キュー211に格納されたWrite要求は、最後のチェックポイント以降に発行されたものであるため、逆にこれを参照して待機系ファイル23から更新前のデータを読み出し、この読み出した更新前のデータを用いて運用系ファイル14をロールバックする。これにより、運用系ファイル14および待機系ファイル23の双方のファイルが最後のチェックポイント時点の状態になる。

【0024】(5) そして、待機系システム20は、未確定キュー211に残存するWrite要求をすべてキャンセルする。

【0025】これにより、チェックポイント時点からの再開始が可能となる。

【0026】次に、この発明の実施の形態を説明する。

【0027】(第1の実施形態) まず、この発明の第1の実施形態を説明する。図2にはこの発明の第1の実施形態に係るコンピュータシステムのシステム構成が示されている。図2に示したように、本実施形態のコンピュータシステムは、コンピュータがプライマリコンピュータ30と、バックアップコンピュータ40とで2重化されており、これらはネットワーク50で接続されている。このプライマリコンピュータ30とバックアップコンピュータ40とは、前述した運用系システム10および待機系システム20双方をそれぞれに備えており、いずれかで運用系システム10が動作するときに、他方では待機系システム20が動作する。ここでは、プライマリコンピュータ30側で運用系システム10、バックアップコンピュータ40側で待機系システム20をそれぞれ説明する。

【0028】プロセス35は、プライマリコンピュータ30上で実行され、プライマリファイル39とバックアップファイル41とで2重化されたファイルを更新する。ここで、プライマリファイル39はプライマリコンピュータ30上に、バックアップファイル41はバックアップコンピュータ40上に配置され、プライマリコンピュータ30上のファイルシステム36およびバックアップコンピュータ40上のファイルシステム48を介して更新される。

【0029】プライマリコンピュータ30上のファイルシステム36は、プライマリファイル操作部38とプライマリファイル復元部37とを含んでいる。一方、バックアップコンピュータ40上のファイルシステム48

は、バックアップファイル操作部43、未確定キュー431、確定キュー432、バックアップファイル更新部44およびプライマリファイル復元情報読み出し部42を含んでいる。

【0030】プロセス35がこの2重化されたファイルを更新する場合、プライマリファイル操作部38およびバックアップファイル操作部43を経由して行なう。プロセス35がこの2重化されたファイルに対応するwriteを行なうと、プライマリファイル39は、そのまま即座に更新されるが、バックアップファイル41はその時点では更新されずに、「ファイル書き込み情報」がバックアップファイル操作部43を経由して、バックアップコンピュータ40上の未確定キュー431に保存される。

【0031】また、プロセス35がチェックポイントを採取する場合には、チェックポイント制御部31が、チェックポイント情報保存部32とプライマリファイル操作部38とにその指示を出す。チェックポイント情報保存部32は、チェックポイント採取の指示を受け取ると、チェックポイント情報(アドレス空間とプロセスコンテキスト)をプライマリコンピュータ30上およびバックアップコンピュータ40上に保存する(プライマリコンピュータ30上のチェックポイント情報34およびバックアップコンピュータ40上のチェックポイント情報45)。

【0032】一方、プライマリファイル操作部38は、チェックポイント採取の指示を受け取ると、バックアップファイル操作部43を経由して、未確定キュー431に保存されていた「ファイル書き込み情報」を確定キュー432に移動させる。この確定キュー432に移動された「ファイル書き込み情報」は、チェックポイント採取後に、バックアップファイル更新部44によってバックアップファイル41の更新のために使用され、バックアップファイル41の更新後に廃棄される。これにより、チェックポイント以降にプライマリファイル39に対して行なわれたものと同じwrite操作が、バックアップファイル41に対しても行なわれることになる。

【0033】プロセス35がアバートなどの障害を発生させ、プロセス35をプライマリコンピュータ30上で最後に採取したチェックポイントから再実行する場合、アドレス空間とプロセスコンテキストとは、プライマリコンピュータ30上のチェックポイント情報復元部37によって復元される。

【0034】ファイルに関しては、バックアップファイル41は、チェックポイント以降の更新は未だ未確定キュー431に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。しかしながら、プライマリファイル39は、チェックポイント以降にすでに更新が行なわれているの

(6)

特開平10-133927

で復元が必要である。したがって、未確定キュー431に保存された「ファイル書き込み情報」に基づき、プライマリファイル39の更新前データをバックアップファイル41からreadし、このreadした更新前データをプライマリファイル39にwriteすることによって復元する。そして、この後、未確定キュー431に保存された「ファイル書き込み情報」を廃棄する。なお、確定キュー432に「ファイル書き込み情報」が保存されている場合には、その「ファイル書き込み情報」のバックアップファイル41への反映が完了した後に、前述した復元処理を開始する。

【0035】一方、プライマリコンピュータ30またはプライマリコンピュータ30を制御するオペレーティングシステムがシステムダウンなどの障害を発生させ、プロセス35をバックアップコンピュータ40上で最後に採取したチェックポイントから再実行する場合には、アドレス空間とプロセッサコンテキストとは、チェックポイント情報復元部46によってプロセス47に復元される。

【0036】ファイルに関しては、バックアップファイル41は、チェックポイント以降の更新は未だ未確定キュー431に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。

【0037】なお、この「ファイル書き込み情報」のプライマリコンピュータ30からバックアップコンピュータ40への転送については最適化が可能である。障害が発生したときに、プライマリコンピュータ30がダウンしなかった場合は、プライマリファイル39を復元し、プライマリファイル39を用いてチェックポイントからの処理を再開する。一方、障害が発生したときに、プライマリコンピュータ30がダウンした場合には、バックアップファイル41を用いてチェックポイントから処理を再開する。

【0038】それゆえに、「ファイル書き込み情報」は、プライマリファイル操作部38からバックアップファイル操作部43に即時に送る必要はない。すなわち、これらの「ファイル書き込み情報」は、次のチェックポイントまでに送ればよいので、転送効率を考慮すると、一旦プライマリファイル操作部38において蓄積しておき、「一定容量蓄積された」、「一定時間経過した」および「チェックポイント採取が要求された」といった事象の発生をトリガとして、バックアップファイル操作部43にまとめて送るとすることが可能である。

【0039】図3には、本実施形態を適用するコンピュータシステムの概略構成が示されている。コンピュータはプライマリコンピュータ30とバックアップコンピュータ40とで2重化されており、プライマリコンピュータ30にはディスク装置60aが、バックアップコンピュータ40にはディスク装置60bがそれぞれ接続され

ている。プロセス35はプライマリコンピュータ上で実行され、また、このプロセス35がアクセスするファイルは、プライマリファイル39とバックアップファイル41とで2重化されており、各々ディスク装置60aとディスク装置60bとに配置されている。

【0040】そして、チェックポイントは、チェックポイント情報をプライマリコンピュータ30側（プライマリチェックポイント情報34）と、バックアップコンピュータ40側（バックアップチェックポイント情報45）の両方に保持する。なお、この図では、チェックポイントをディスク装置上に保持しているが、メモリ上に保持しても構わない。

【0041】もし、プライマリコンピュータ30またはプライマリコンピュータ30を制御するオペレーティングシステムにシステムダウンなどの障害が発生した場合には、バックアップコンピュータ40側でチェックポイント情報45を用いてプロセス47を再実行する。この場合プロセス47は、バックアップファイル41を使用することになる。

【0042】また、プライマリファイル39またはバックアップファイル41を複数個持ち、3重化以上のファイルシステムを作ることにも可能である。この場合、たとえば3重化ファイルシステムならば、

(1) 2個のプライマリファイルと1個のバックアップファイル

(2) 1個のプライマリファイルと2個のバックアップファイル

といった組み合わせが考えられる。

【0043】図4は、本実施形態においてファイルを更新する様子を示す図である。この例では、プライマリコンピュータ30上で動くプロセス35が、4バイトのデータ“ABCD”を持つ2重化されたファイル（プライマリコンピュータ30上のプライマリファイル39と、バックアップコンピュータ40上のバックアップファイル41）に対し、時刻t1において1バイト目に“X”をwriteしている(1)。これによってプライマリファイル39は即時に更新されるが、バックアップファイル41は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【0044】この後、時刻t2においてチェックポイントが採取されることによって、先程の「ファイル書き込み情報」の実行が確定する(2)。そして時刻t2以降で、確定された「ファイル書き込み情報」に基づいて、バックアップファイル41の更新を実行している。

【0045】図5は、本実施形態において障害発生時にプライマリファイルを復元する様子を示す図である。この例では、プライマリコンピュータ30上で動くプロセス35が、4バイトのデータ“ABCD”を持つ2重化されたファイル（プライマリコンピュータ30上のプライマリファイル39と、バックアップコンピュータ40

(7)

特開平10-133927

上のバックアップファイル41)に対し、時刻t1において1バイト目に“X”をwriteしている(1)。これによってプライマリファイル39は即時に更新されるが、バックアップファイル41は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【0046】この後、時刻t2において障害が発生している(2)。すなわち、時刻t1における「ファイル書き込み情報」でプライマリファイル39は更新されているため復元の必要があるが、バックアップファイル41は未だ更新されていないため復元の必要がない。ここで時刻t1において保存された「ファイル書き込み情報」によって、プライマリファイル39の更新部分がかわる。そこで、プライマリファイル39の復元においては、未確定の「ファイル書き込み情報」に示された位置のデータをバックアップファイル41からreadし、そのreadしたデータをプライマリファイル39にwriteすることによって、プライマリファイル39を復元する。

【0047】そして、プライマリコンピュータ30上で取られているチェックポイントを用いて、プライマリコンピュータ30上でプロセス35を再実行している。この再実行されたプロセス35は、復元されたプライマリファイル39を使用する。

【0048】図6は、ファイル操作部が「ファイル書き込み」を指示されたときの処理の流れを示すフローチャートである。この場合、まず、「ファイル書き込み情報」を保存し、未確定キュー431にリンクする(ステップA1)。次に、「ファイル書き込み情報」にしたがって、プライマリファイル39の更新を行なう(ステップA2)。この時点で、「ファイル書き込み」操作は完了したとして、要求側に完了通知を行なう(ステップA3)。

【0049】図7は、ファイル操作部が「チェックポイント採取」を指示されたときの処理の流れを示すフローチャートである。この場合、保存されている「ファイル書き込み情報」を未確定キュー431から確定キュー432に移動する(ステップB1)。

【0050】図8は、バックアップファイル更新部の処理の流れを示すフローチャートである。この場合、まず、確定キュー432に「ファイル書き込み情報」がリンクされているかどうかを検査する(ステップC1)。もし、リンクされていない場合(ステップC1のN)、バックアップファイル更新部44は、この検査を続行する。一方、リンクされている場合には(ステップC1のY)、確定キュー432にリンクされている「ファイル書き込み情報」に基づいて、バックアップファイル41を更新する(ステップC2)。そして、実行した「ファイル書き込み情報」を確定キュー432からはずす(ステップC3)。

【0051】図9は、プロセス35にアボートなどの障

害が発生し、プロセス35をプライマリコンピュータ30上で最後に採取したチェックポイントから再実行する場合の処理の流れを示すフローチャートである。

【0052】プロセス35に障害が発生すると、まず、プライマリコンピュータ30上のチェックポイント情報復元部33に、「アドレス空間とプロセッサコンテキストとの復元を指示する(ステップD1)。次に、プライマリファイル復元部33に、「プライマリファイルの復元」を指示する(ステップD2)。

【0053】図10は、プライマリコンピュータ30上のチェックポイント情報復元部が「アドレス空間とプロセッサコンテキストの復元」を指示された場合の処理の流れを示すフローチャートである。この場合、まず、プロセス35のアドレス空間を復元する(ステップE1)。次に、プロセス35のチェックポイント採取時のプロセッサコンテキストの状態を復元する(ステップE2)。

【0054】図11は、プライマリファイル復元部37が、「プライマリファイルの復元」を指示された場合の処理の流れを示すフローチャートである。この場合、まず、未確定キュー431に、「ファイル書き込み情報」がリンクされているかどうかを検査する(ステップF1)。「ファイル書き込み情報」がリンクされている場合には(ステップF1のY)未確定キュー431にリンクされている「ファイル書き込み情報」にしたがって、プライマリファイル39の中の更新されている部分のデータをバックアップファイル41からreadし、そのReadしたデータをプライマリファイル39にwriteすることにより、プライマリファイル39のその更新されている部分のデータを復元する(ステップF2)。そして、復元に利用した「ファイル書き込み情報」を、未確定キュー431からはずす(廃棄する)(ステップF3)。この処理は、未確定キュー431にリンクした「ファイル書き込み情報」が無くなるまで繰り返される。

【0055】プライマリコンピュータ30またはプライマリコンピュータ30を制御するオペレーティングシステムにシステムダウンなどの障害が発生した場合には、プロセス35をバックアップコンピュータ40上で最後に採取したチェックポイントから再実行する。この場合は、バックアップファイル41で処理を引き継ぐ。図12は、障害が発生したときに、バックアップファイル41で処理を引き継ぐ様子を示す図である。

【0056】この例では、プライマリコンピュータ30上で動作するプロセス35が、4バイトのデータ“ABCD”を持つ2重化されたファイル(プライマリコンピュータ30上のプライマリファイル39と、バックアップコンピュータ40上のバックアップファイル41)に対し、時刻t1において1バイト目に“X”をwriteしている(1)。これによってプライマリファイル3



(8)

特開平10-133927

9は即時に更新されるが、バックアップファイル41は即時には更新されずに、「ファイル書き込み情報」のみを保存している。

【0057】この後、時刻t2においてプライマリコンピュータ30に障害が発生している(2)。この場合、バックアップコンピュータ40上に取られたチェックポイントを用いて、バックアップコンピュータ40上でプロセス47を再実行している。このとき、プロセス47は、バックアップファイル41を用いて処理を継続するわけだが、時刻t1においてプライマリファイル39は更新されているが、バックアップファイル41は未だ更新されていないので、バックアップコンピュータ40上でのプロセス47の再実行においては、バックアップファイル42がそのまま使用できる。

【0058】なお、障害発生によりバックアップファイルを切り離した場合には、その後に新たなバックアップファイルを作成することによって、再び図1の様な初期状態を再現することができ、再度の障害発生に対しても回復処理が可能となる。

【0059】また、障害発生によってバックアップファイルで処理を引き継ぎ、チェックポイントから処理を再実行した場合には、その後、バックアップファイルをプライマリファイルとして新たなバックアップファイルを作成することにより、再び図1の様な初期状態を再現することができ、再度の障害発生に対しても回復処理が可能となる。この再度バックアップファイルを作成する場合には、以下の様な2つの方法がある。

【0060】(1)バックアップファイル切り離し後のプライマリファイルの更新情報とデータとを保存しておき、バックアップファイルを再接続する場合には、バックアップファイルに前記切り離し後のプライマリファイルの更新情報とデータとを反映させる。

【0061】(2)プライマリファイルをバックアップファイルにコピーする。ただし、コピー中にもプライマリファイルが更新され続けている場合には、コピーを始めると同時にファイルの更新情報とデータとをバックアップファイルにも反映させる。

【0062】さらに、この2つの方法を組み合わせた以下の様な方法も有効である。

【0063】(3)切り離されたバックアップファイル(あるいは障害発生前のプライマリファイル)を再接続することを前提に、一定時間が経過するまでは(1)の方法が取れる様に、バックアップファイル切り離し後のプライマリファイルの更新情報とデータとを保存しておく。一定時間を経過したら、(1)の方法は締め、バックアップファイル切り離し後のプライマリファイルの更新情報とデータとの保存は止めて、(2)の方法を取るようになる。また、切り離されたバックアップファイル以外のファイルで再接続する場合にも、バックアップファイル切り離し後のプライマリファイルの更新情報とデ

ータとの保存は止めて、(2)の方法を取る。

【0064】(第2の実施形態)次に、この発明の第2の実施形態を説明する。第1の実施形態では、2重化されたコンピュータシステムを説明したが、この発明は、2重化されていないコンピュータ上のファイルシステムに適用しても効果がある。そこで、本実施形態では、2重化されていないコンピュータ上のファイルシステムに適用した場合を例に説明する。図13は、この発明を2重化されていないコンピュータ上のファイルシステムに適用した場合の構成図である。このシステムでは、コンピュータは2重化されておらず、コンピュータ30のみが存在する。プロセス35は、このコンピュータ30上で実行され、プライマリファイル39とバックアップファイル41とで2重化されたファイルを更新する。すなわち、これらプライマリファイル39およびバックアップファイル41は、共にコンピュータ30上に配置され、ファイルシステム36を介して更新される。

【0065】コンピュータ30上のファイルシステム36は、プライマリファイル操作部38、プライマリファイル復元部37、バックアップファイル操作部43、未確定キュー431、確定キュー432、バックアップファイル更新部44およびプライマリファイル復元情報読み出し部42を含んでいる。

【0066】プロセス35がこの2重化されたファイルを更新するときは、プライマリファイル操作部38およびバックアップファイル操作部43を経由して行なう。プロセス35がこの2重化されたファイルに対するwriteを行なうと、プライマリファイル39はそのまま更新されるが、バックアップファイル41は更新されずに、「ファイル書き込み情報」がバックアップファイル操作部43を経由して未確定キュー431に保存される。

【0067】また、プロセス35がチェックポイントを採取するときには、チェックポイント制御部31が、チェックポイント情報保存部32とプライマリファイル操作部43に指示を出す。チェックポイント情報保存部32はチェックポイント採取の指示を受けると、アドレス空間とプロセッサコンテキストとをコンピュータ30上に行なう(チェックポイント情報34)。

【0068】一方、プライマリファイル操作部38は、チェックポイント採取の指示を受けると、バックアップファイル操作部43を経由して、未確定キュー431に保存されていた「ファイル書き込み情報」を確定キュー432に移動させる。確定キュー432に移動された「ファイル書き込み情報」は、チェックポイント採取後に、バックアップファイル更新部44によってバックアップファイル41の更新のために使用され、バックアップファイル41の更新後に廃棄される。これにより、チェックポイント以降にプライマリファイル39に対して行なわれたのと同じように、write操作がバックア

(9)

特開平10-133927

ップファイル41に対して行なわれる。

【0069】プロセス35にアボートなどの障害が発生し、プロセス35をコンピュータ30上で最後に採取したチェックポイントから再実行する場合、アドレス空間とプロセッサコンテキストは、コンピュータ30上のチェックポイント情報復元部33によって復元される。

【0070】ファイルに関しては、バックアップファイル41は、チェックポイント以降の更新が未だ未確定キュー431に「ファイル書き込み情報」が保存されているだけであり、実際には更新されていないので復元は不要である。しかしながら、プライマリファイル39は、チェックポイント以降にすでに更新が行なわれているので復元が必要である。したがって、未確定キュー431に保存された「ファイル書き込み情報」に基づき、プライマリファイル39の更新前データをバックアップファイル41からreadし、このReadした更新前データをプライマリファイル39にwriteすることによって復元する。そして、この後、未確定キュー431に保存された「ファイル書き込み情報」を廃棄する。なお、確定キュー432に「ファイル書き込み情報」が保存されている場合には、その「ファイル書き込み情報」のバックアップファイル41への反映が完了した後に、前述した復元処理を開始する。

【0071】図14には、本実施形態を適用するコンピュータシステムの概略構成が示されている。本実施形態のシステムはコンピュータ30のみで稼働し2重化されていない。コンピュータ30にはディスク装置60aとディスク装置60bとが接続されている。プロセス35はコンピュータ30上で実行され、また、このプロセス35がアクセスするファイルは、プライマリファイル39とバックアップファイル41とで2重化されており、各々ディスク装置60aとディスク装置60bとに配置されている。

【0072】このように、この発明を適用することにより、プロセスのアドレス空間やプロセッサのコンテキストなどの状態(チェックポイント情報)を定期的に保存しながら実行を続け、障害が発生したときには最後に保存したチェックポイントからプロセスを再実行させることによる障害時対策を施したシステムにおいて、ファイルの更新を行なう際に、一旦更新前データをファイルから読み込む必要がなくなるため、ファイルの更新性能が大幅に改善される。

【0073】なお、前述の実施形態に記載したファイルの管理方法は、コンピュータに実行させることのできるプログラムとしてフロッピーディスク、光ディスクおよび半導体メモリなどの記録媒体に格納して頒布することが可能である。

【0074】

【発明の効果】以上詳述したように、この発明によれば、プロセスがファイルの更新を要求したときに、その

更新内容を示す更新情報を取得して保存するとともにプライマリファイルのみを即座に更新し、チェックポイントが採取された後に、その保存しておいた更新情報で示される更新内容をバックアップファイルに反映させる。そして、たとえばプロセスがアボートしたときなどには、保存しておいた更新情報に基づいて、最後に採取したチェックポイント以降に更新されたデータに対応する更新前のデータをバックアップファイルからすべて読み出し、この読み出した更新前のデータを用いてプライマリファイルをチェックポイント時点に復元し、プロセスの再実行を開始する(バックアップファイルを用いたプロセスの再実行の開始も可能)。

【0075】すなわち、このコンピュータシステムにおいては、従来のようにファイルを更新するときに、更新前のデータを読み出して退避させておくといった処理の完了を通常処理に待機させることなく、障害時のファイルのリカバリが実現されることになり、信頼性を損なうことなくファイルの更新性能を飛躍的に向上させることが可能となる。

【図面の簡単な説明】

【図1】この発明の基本原理解を説明するための概念図。

【図2】この発明の第1の実施形態に係るコンピュータシステムのシステム構成を示す図。

【図3】同実施形態を適用するコンピュータシステムの概略構成を示す図。

【図4】同実施形態においてファイルを更新する様子を示す図。

【図5】同実施形態において障害発生時にプライマリファイルを復元する様子を示す図。

【図6】同実施形態のファイル操作部が「ファイル書き込み」を指示されたときの処理の流れを示すフローチャート。

【図7】同実施形態のファイル操作部が「チェックポイント採取」を指示されたときの処理の流れを示すフローチャート。

【図8】同実施形態のバックアップファイル更新部の処理の流れを示すフローチャート。

【図9】同実施形態のプロセスにアボートなどの障害が発生し、プロセスをプライマリコンピュータ30上で最後に採取したチェックポイントから再実行する場合の処理の流れを示すフローチャート。

【図10】同実施形態のプライマリコンピュータ上のチェックポイント情報復元部が「アドレス空間とプロセッサコンテキストとの復元」を指示された場合の処理の流れを示すフローチャート。

【図11】同実施形態のプライマリファイル復元部が「プライマリファイルの復元」を指示された場合の処理の流れを示すフローチャート。

【図12】同実施形態の障害が発生したときにバックアップファイルで処理を引き継ぐ様子を示す図。

(10)

特開平10-133927

【図13】この発明の第2の実施形態に係るコンピュータシステムのシステム構成を示す図。

【図14】同実施形態を適用するコンピュータシステムの概略構成を示す図。

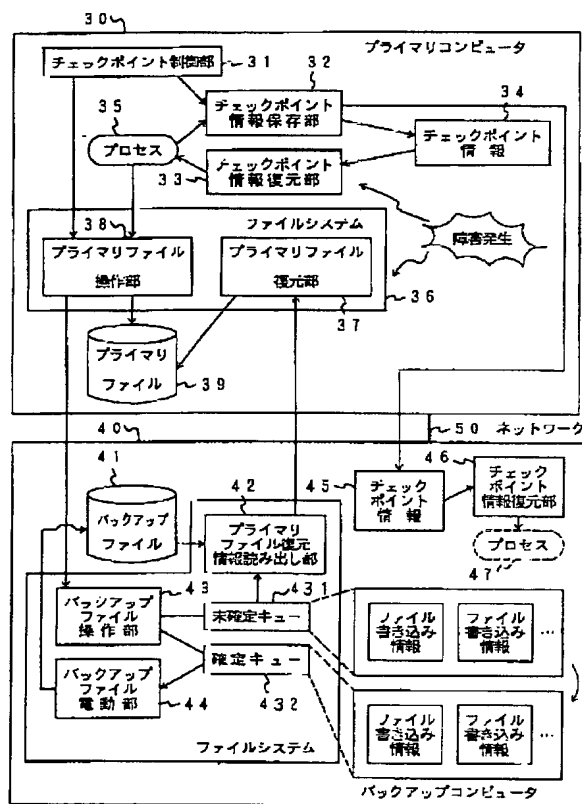
【図15】従前のファイルに対する書き込みをキャンセルすることが困難なため、ファイルに対して書き込みを行なうときに、データをファイルに書き込む前に書き込み以前のデータを事前に読み込んで保存を行ない、その後、ファイルへのデータ書き込みを行なう従来のシステムの仕組みを説明する図。

【符号の説明】

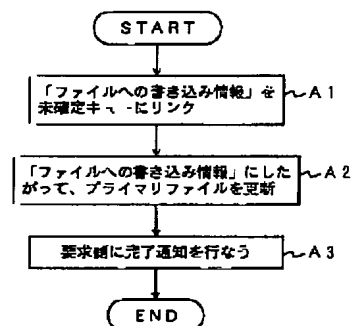
10…運用系システム、11…アプリケーションプログラム、12…ジャケットルーチン、13…OSバッファキャッシュ、14…ディスク装置、20…待機系システム、21デーモン、211…未確定キュー、212…確

定キュー、22…OSバッファキャッシュ、23…ディスク装置、30…プライマリコンピュータ、31…チェックポイント制御部、32…チェックポイント情報保存部、33…チェックポイント情報復元部、34…チェックポイント情報、35…プロセス、36…ファイルシステム、37…プライマリファイル復元部、38…プライマリファイル操作部、39…プライマリファイル、40…バックアップコンピュータ、41…バックアップファイル、42…プライマリファイル復元情報読み出し部、43…バックアップファイル操作部、431…未確定キュー、432…確定キュー、44…バックアップファイル更新部、45…チェックポイント情報、46…チェックポイント情報復元部、47…プロセス、50…ネットワーク、60a、60b…ディスク装置。

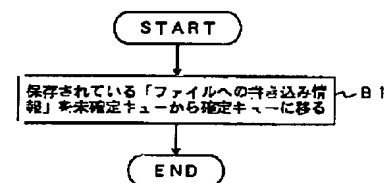
【図2】



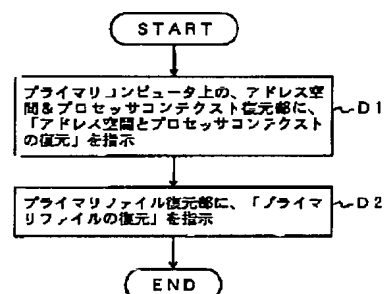
【図6】



【図7】



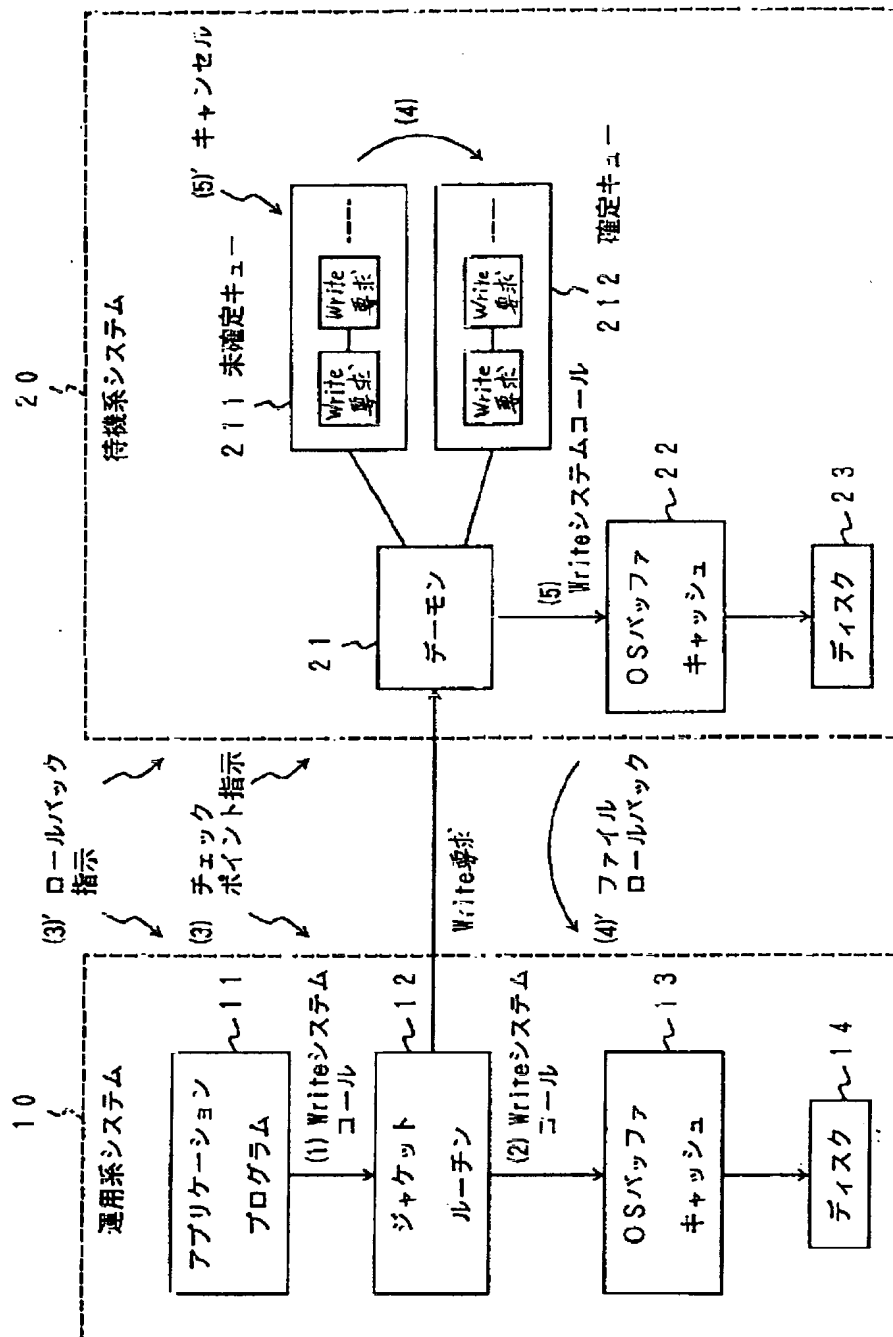
【図9】



(11)

特開平10-133927

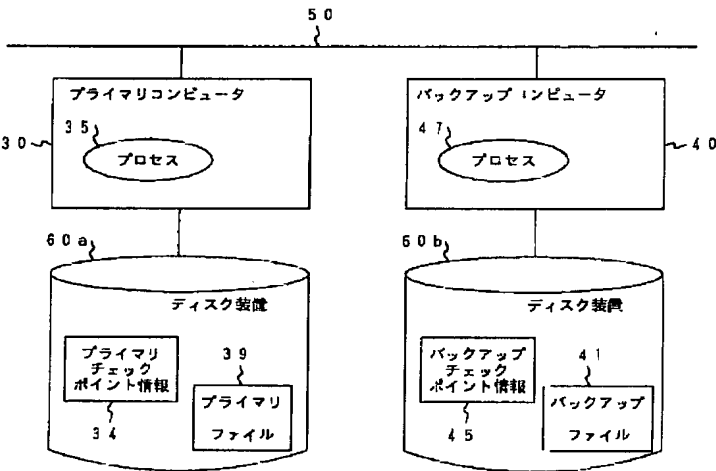
【図1】



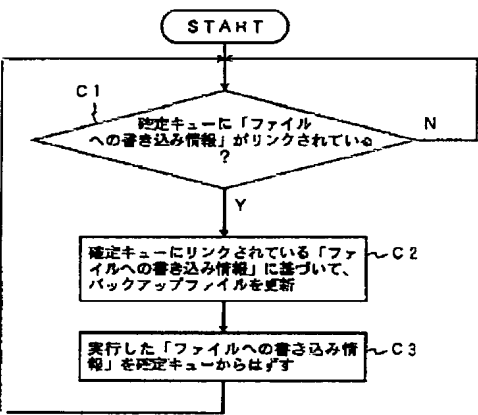
(12)

特開平10-133927

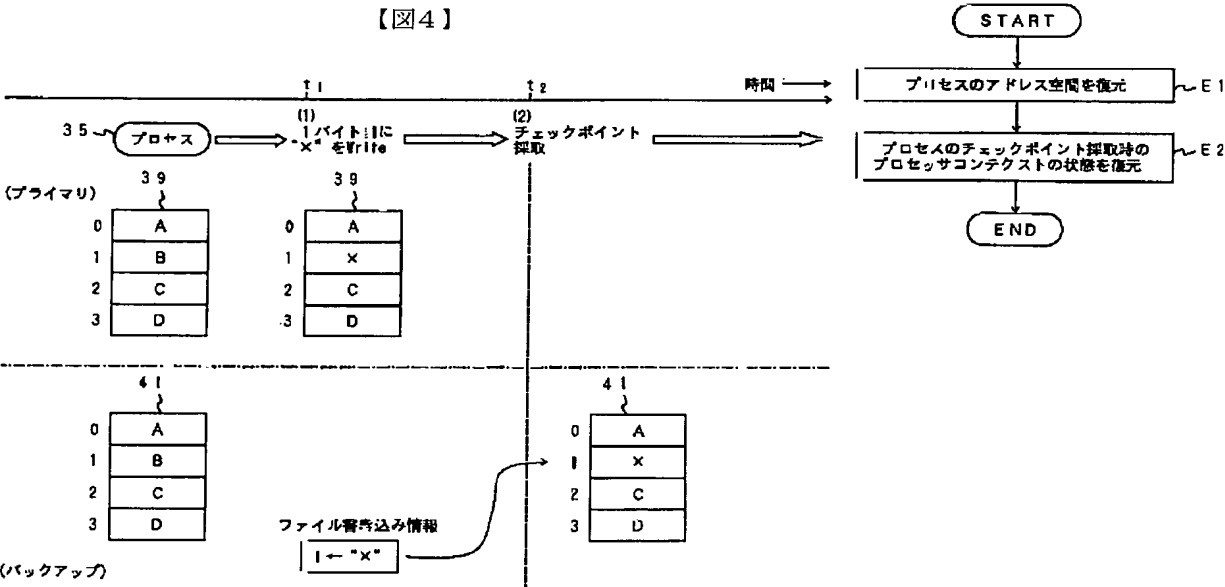
【図3】



【図8】



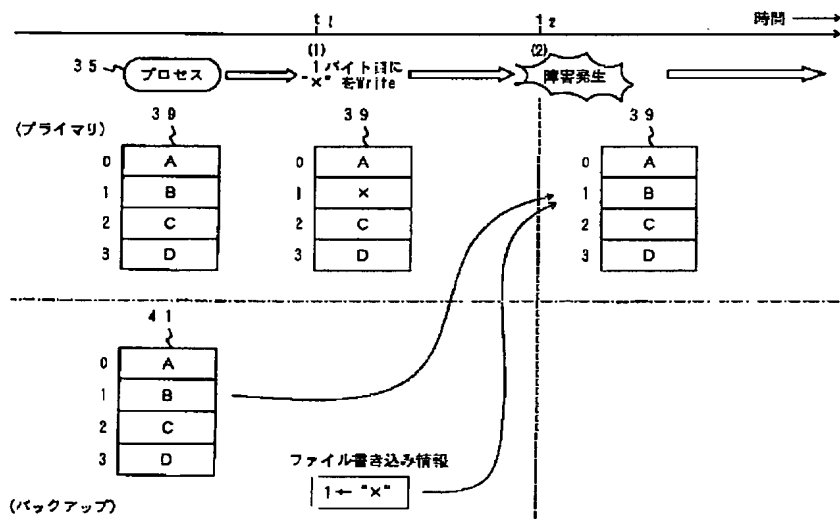
【図10】



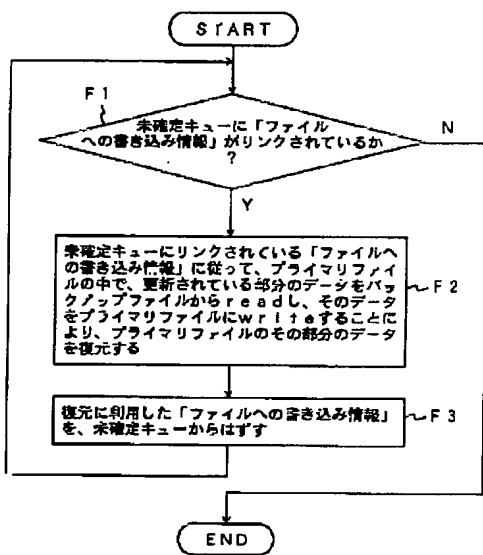
(13)

特開平10-133927

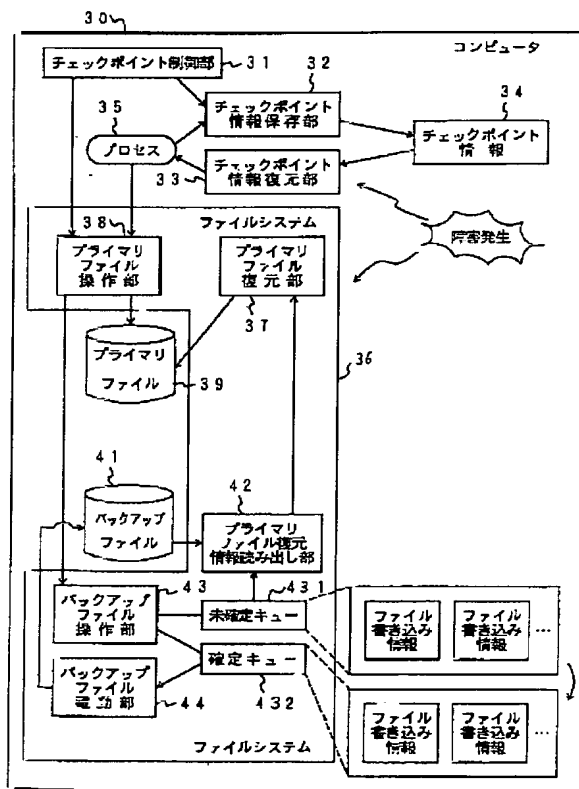
【図5】



【図11】



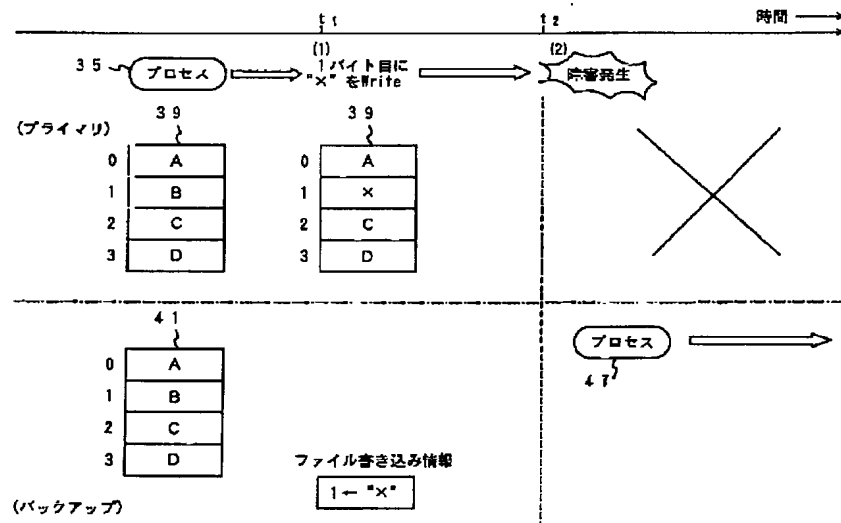
【図13】



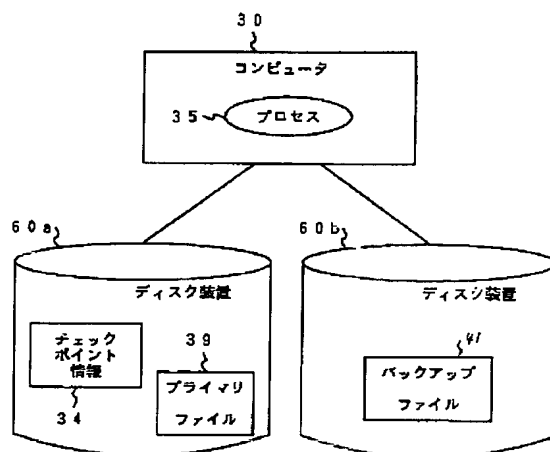
(14)

特開平10-133927

【図12】



【図14】



(15)

特開平10-133927

【図15】

